# 传输层协议

传输层提供端到端的进程间通信，传统的TCP/IP协议传输层提供传输控制协议TCP和用户数据报协议UDP。流控制传输协议SCTP是随着多媒体发展而出现的传输层协议。

## 1 传输控制协议TCP

TCP是面向字节流的协议，发送方与接收方以字节流的形式按顺序收发数据；TCP是面向连接的协议，优点是是可靠，缺点是传输速度慢，开销大；通信前需进行三次握手，结束后进行四次挥；TCP是点对点通信，一条TCP连接只能连两个点；提供全双工通信，允许通信双方任何时候都能收发数据；TCP提供可靠的传输，具有无差错、不丢失、不重复的特点。

**1.1 TCP数据段格式**

TCP是面向字节流的，通过TCP传送的字节流中的每个字节都按顺序编号，而报头中的Sequence Number字段值则指的是本报文段数据的第一个字节的序号。Acknowledgment Number是期望收到对方下个报文段的第一个数据字节的序号。  
 Offset：占4位，指TCP报文段的报头长度，包括固定的20字节和TC POptions字段。  
 Reserved：占6位，保留为今后使用，目前为0。  
 TCP flags的C、E、U、A、P、R、S、F字段用来说明该报文的性质。意义如下：  
 C(CWR)和E(ECE)用来支持ECN(显示阻塞通告)。  
 U(URGENT)：当 URG=1时，它告诉系统此报文中有紧急数据，应优先传送(比如紧急关闭)，这要与紧急指针字段配合使用。  
 A(ACK)：仅当 ACK=1时确认号字段才有效。建立 TCP 连接后，所有报文段都必须把 ACK 字段置为 1。  
 P(PUSH)：若TCP连接的一端希望另一端立即响应，PSH字段便可以“催促”对方，不再等到缓存区填满才发送。  
 R(RESET)：若 TCP 连接出现严重差错，RST 置为 1，断开 TCP 连接，再重新建立连接。  
 S(SYN)：用于建立和释放连接，当SYN=1时，表示建立连接。  
 F(FIN)：用于释放连接，当 FIN=1，表明发送方已经发送完毕，要求释放TCP 连接。  
 Window：占2个字节。窗口值是指发送者自己的接收窗口大小，因为接收缓存的空间有限。  
 CheckSum：占2个字节。和UDP报文一样，有一个检验和，用于检查报文是否在传输过程中出差错。  
 Urgent Pointer：占2字节。当URG=1时才有效，指出本报文段紧急数据的字节数。

**1.2 TCP连接的建立**

在TCP／IP中，TCP协议通过三次握手来建立连接，从而提供可靠的连接服务。

第一次握手：发送端首先发送一个带SYN（synchronize）标志的数据包给接收方（第一次的seq序列号是随机产生的，这样是为了网络安全，如果不是随机产生初始序列号，黑客将会以很容易的方式获取到你与其他主机之间的初始化序列号，并且伪造序列号进行攻击）；

第二次握手：接收端收到后，回传一个带有SYN/ACK（acknowledgement）标志的数据包以示传达确认信息（SYN是为了告诉发送端，发送方到接收方的通道没问题；ACK 用来验证接收方到发送方的通道没问题）；

第三次握手：SYN＋ACK包，客户端收到服务器端发来的确认包ACK（ACK＝k＋1），来发送这个包来发送，客户端和服务器端进入建立状态，完成三次握手。

TCP四次挥手：

主动断开方（客户端/服务端）发送一个FIN，用来关闭主动断开方（客户端/服务端）到被动断开方（客户端/服务端）的数据传送；  
 被动断开方（客户端/服务端）收到这个FIN，它发回一 个ACK，确认序号为收到的序号加1。和SYN一样，一个FIN将占用一个序号  
 被动点开方（客户端/服务端）关闭与主动断开方（客户端/服务端）的连接，发送一个FIN给主动断开方（客户端/服务端）；  
 主动断开方（客户端/服务端）发回ACK报文确认，并将确认序号设置为收到序号加1。

**1.3 TCP流量控制、拥塞控制和差错控制**

由于接收方缓存的限制，发送窗口不能大于接收方接收窗口。在报文段首部有一个字段就叫做窗口(rwnd)，这便是用于告诉对方自己的接收窗口，窗口的大小是可以变化的。

TCP的流量和阻塞控制采用“慢启动”、“加性增”、“乘性减”的策略。  
　 慢启动：初始的窗口值很小，但是按指数规律渐渐增长，直到达到慢开始门限(ssthresh)。  
　 加性增：窗口值达到慢开始门限后，每发送一个报文段，窗口值增加一个单位量。  
　 乘性减：无论什么阶段，只要出现超时，则把窗口值减小一半。

## 2 UDP用户数据报协议

UDP是面向报文的协议，只对IP数据报做简单封装，减少了报头的开销；UDP是无连接的，发送数据前不需要建立连接，即在发送端，应用层将数据传递给传输层的 UDP 协议，UDP 只会给数据增加一个 UDP 头标识下是 UDP 协议，然后就传递给网络层了，在接收端，网络层将数据传递给传输层，UDP 只去除 IP 报文头就传递给应用层，不会任何拼接操作，这使得数据传输速度高，UDP花费的开销小也比TCP建立连接的小，同时也体现了不稳定性和不可靠性；UDP不提供流量控制，没有拥塞机制，阻塞时丢弃数据，不延时；UDP 不止支持一对一的传输方式，同样支持一对多，多对多，多对一的方式，也就是说 UDP 提供了单播，多播，广播的功能；UDP的传输机制决定了其最大的优点：高效率。

## 3 流控制传输协议SCTP

SCTP是以数据块为单位传输的；SCTP是面向连接的，两端都可以绑定到多个IP上，只要有其中一对IP能通，这条SCTP连接就还可以用；一条SCTP连接里面，可以区分多条不同的流（stream），不同的流之间的数据传输互不干扰；在同一条stream里面，SCTP支持有序/无序两种传输方式，应用程序在调用sendmsg()的时候，需要指定用哪一条stream传输，以及指定这条要发送的消息是需要有序传输还是无序传输的。如果在传输过程中丢包，则有序传递模式可能会在接收端被阻塞，而无序传输模式不会在接收端被阻塞；SCTP的建立过程需要四步握手，server端在收到连接请求时，不会立即分配内存缓存起来，而是返回一个COOKIE。client端需要回送这个COOKIE，server端校验之后，从cookie中重新获取有效信息（比如对端地址列表），才会最终建立这条连接。这样，可以避免类似TCP的SYN攻击；SCTP协议本身有heartbeat机制来监控连接/路径的可用性。